

# Lightweight Automatic Generation of Traceability Information

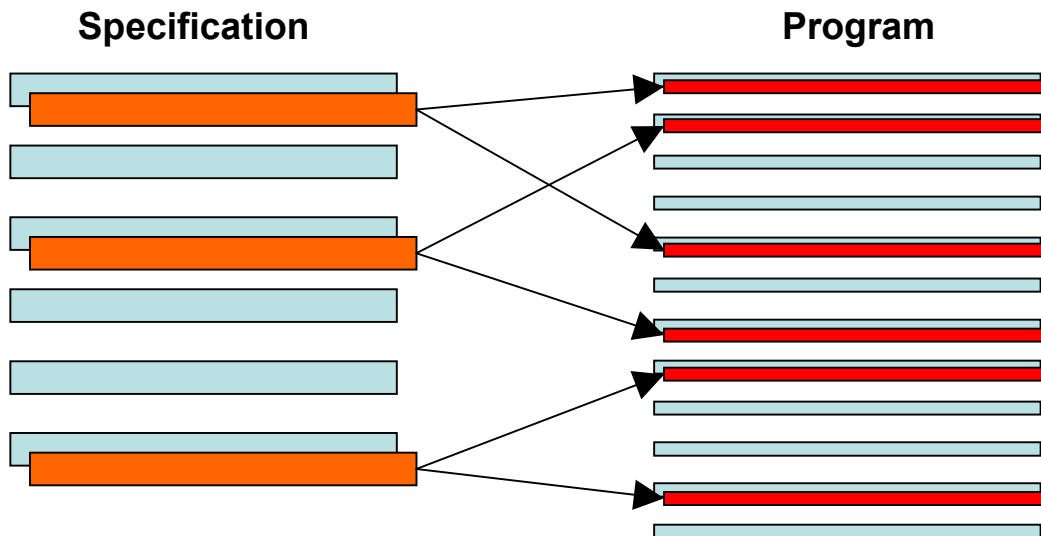
## PROBLEM

Traceability relations between requirements and code are generally derived manually, and must be manually updated when software or requirements change, requiring effort and expertise. The effort can be reduced when code is automatically generated, but this requires substantial modification to the code generator.

## SOLUTION

A general technique for deriving traceability relations for automatically generated code, treating the code generator as a black box. The technique can also be applied to other kinds of automatic generation.

## TECHNOLOGY



Synthesis generates lines of program code (blue, right) from lines of specification (blue, left). Technique detects synthesized program changes (red, right) which result from specification changes (orange, left). Arrows denote derived traceability links.

# Explanation of Accomplishment

- **POC:** Julian Richardson (ASE Group, Code IC, [julianr@email.arc.nasa.gov](mailto:julianr@email.arc.nasa.gov))
- **Work funded by:** ITSr Certifiable Program Synthesis.
- **Background:** Tracing from a program's requirements to the parts of the program which implement those requirements is important – by linking related artifacts in a software development, it can help to ensure that consistency is maintained as the software evolves, and help to pinpoint software errors. Traceability is required by the avionics standard DO-178B. Programmers input traceability information manually into tools such as DOORS. This information is hard to derive, requiring understanding of both problem domain and program, and hard to maintain when the program or requirements change. Synthesis systems and compilers can be extended to derive traceability information in parallel with the synthesis/compilation process, but this has previously required extensive modification of the entire synthesis system.
- **Accomplishment:** We have developed a new, very lightweight technique for deriving traceability information which works with any kind of automatic generation process. The technique observes the differences induced in the synthesized program when small changes are made to its specification. Lines of the synthesized program are annotated with the lines of the specification which affected them. The technique can be implemented in less than 100 lines of code. We have demonstrated its flexibility by applying it both to program synthesis (*AutoFilter – input: specifications, output: C code*), and to compilation (*GCC – input: C code, output: assembler code*). The results were presented at the 2<sup>nd</sup> International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE'03) October 7, 2003.
- **Benefits:** We enable traceability information to be derived from any automatic (program) generation process without needing substantial modification or understanding of the generating system.
- **Future Work:** The technique currently relies on the user to provide the small changes which are made to the input specification. We will examine possibilities for automating generation of these changes. In the case that several artifacts are generated from a single specification, the technique may allow deriving linkages between those artifacts. For example, we may be able to tie generated program statements to generated proof obligations during program certification. Currently, the derived traceability information maps lines of generated code to corresponding lines of specification. We will investigate partitioning lines of specification into individual requirements so that we can trace directly to requirements rather than to individual specification lines.